```
UUU         UUU  EEEEEEEEEEEEEEE  TTTTTTTTTTTTTTT  PPPPPPPPPPPP       SSSSSSSSSSSS  YYY         YYY
UUU         UUU  EEEEEEEEEEEEEEE  TTTTTTTTTTTTTTT  PPPPPPPPPPPP       SSSSSSSSSSSS  YYY         YYY
UUU         UUU  EEEEEEEEEEEEEEE  TTTTTTTTTTTTTTT  PPPPPPPPPPPP       SSSSSSSSSSSS  YYY         YYY
UUU         UUU  EEE                   TTT        PPP         PPP  SSS             YYY         YYY
UUU         UUU  EEE                   TTT        PPP         PPP  SSS             YYY         YYY
UUU         UUU  EEE                   TTT        PPP         PPP  SSS             YYY         YYY
UUU         UUU  EEE                   TTT        PPP         PPP  SSS                YYY   YYY
UUU         UUU  EEE                   TTT        PPP         PPP  SSS                YYY   YYY
UUU         UUU  EEE                   TTT        PPP         PPP  SSS                YYY   YYY
UUU         UUU  EEEEEEEEEEE           TTT        PPPPPPPPPPPP       SSSSSSSSS           YYY
UUU         UUU  EEEEEEEEEEE           TTT        PPPPPPPPPPPP       SSSSSSSSS           YYY
UUU         UUU  EEEEEEEEEEE           TTT        PPPPPPPPPPPP       SSSSSSSSS           YYY
UUU         UUU  EEE                   TTT        PPP                        SSS        YYY
UUU         UUU  EEE                   TTT        PPP                        SSS        YYY
UUU         UUU  EEE                   TTT        PPP                        SSS        YYY
UUU         UUU  EEE                   TTT        PPP                        SSS        YYY
UUU         UUU  EEE                   TTT        PPP                        SSS        YYY
UUU         UUU  EEE                   TTT        PPP                        SSS        YYY
UUUUUUUUUUUUUUU  EEEEEEEEEEEEEEE       TTT        PPP             SSSSSSSSSSSS          YYY
UUUUUUUUUUUUUUU  EEEEEEEEEEEEEEE       TTT        PPP             SSSSSSSSSSSS          YYY
UUUUUUUUUUUUUUU  EEEEEEEEEEEEEEE       TTT        PPP             SSSSSSSSSSSS          YYY
```

```
  SSSSSSSS    AAAAAA    TTTTTTTTTT    SSSSSSSS    SSSSSSSS    SSSSSSSS    999999        11
  SSSSSSSS    AAAAAA    TTTTTTTTTT    SSSSSSSS    SSSSSSSS    SSSSSSSS    999999        11
SS          AA      AA      TT      SS          SS          SS          99      99    1111
SS          AA      AA      TT      SS          SS          SS          99      99    1111
SS          AA      AA      TT      SS          SS          SS          99      99      11
SS          AA      AA      TT      SS          SS          SS          99      99      11
  SSSSSS    AA      AA      TT        SSSSSS      SSSSSS      SSSSSS      99999999      11
  SSSSSS    AA      AA      TT        SSSSSS      SSSSSS      SSSSSS      99999999      11
        SS  AAAAAAAAAA      TT              SS          SS          SS          99      11
        SS  AAAAAAAAAA      TT              SS          SS          SS          99      11
        SS  AA      AA      TT              SS          SS          SS          99      11          ....
        SS  AA      AA      TT              SS          SS          SS          99      11          ....
SSSSSSSS    AA      AA      TT      SSSSSSSS    SSSSSSSS    SSSSSSSS    999999      111111        ....
SSSSSSSS    AA      AA      TT      SSSSSSSS    SSSSSSSS    SSSSSSSS    999999      111111        ....
```

```
LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

Table of contents

(1)	54	DECLARATIONS
(1)	97	CONDITION TABLES
(1)	125	TM_SETUP, TM_CLEANUP
(1)	188	CONDITION SUBROUTINES - SETUP AND CLEANUP
(1)	258	FORM_CONDS
(1)	351	VERIFY
(1)	513	VFY_CLEANUP

```
0000          .TITLE  SATSSS91 SATS SYSTEM SERVICE TESTS  $ADJSTK (SUCC S.C.)
0000          .IDENT  'V04-000'
0000
0000    ;
0000    ;*******************************************************************************
0000    ;*                                                                             *
0000    ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                    *
0000    ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                     *
0000    ;*  ALL RIGHTS RESERVED.                                                       *
0000    ;*                                                                             *
0000    ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED      *
0000    ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE     *
0000    ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER      *
0000    ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY      *
0000    ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY      *
0000    ;*  TRANSFERRED.                                                               *
0000    ;*                                                                             *
0000    ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE      *
0000    ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT     *
0000    ;*  CORPORATION.                                                               *
0000    ;*                                                                             *
0000    ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS      *
0000    ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                    *
0000    ;*                                                                             *
0000    ;*                                                                             *
0000    ;*******************************************************************************
0000    ;
0000
0000    ;++
0000    ; FACILITY:     SYSTST (SATS SYSTEM SERVICE TESTS)
0000    ;
0000    ; ABSTRACT:
0000    ;
0000    ;               THIS MODULE CONTAINS SUBROUTINES WHICH, WHEN LINKED
0000    ; WITH SUCCOMMON.OBJ, FORM TEST MODULE SATSSS91 TO TEST SUCCESSFUL
0000    ; OPERATION OF THE $ADJSTK SYSTEM SERVICE. THE SERVICE IS INVOKED
0000    ; UNDER VARIOUS INPUT CONDITIONS WITH VARYING INPUT PARAMETERS. ONLY
0000    ; SUCCESSFUL STATUS CODES ARE EXPECTED IN THIS TEST MODULE.  CORRECT
0000    ; OPERATION OF THE SERVICE FOR EACH OF ITS ISSUANCES IS VERIFIED BY
0000    ; CHECKING FOR AN SS$_NORMAL STATUS CODE, EXPECTED RETURN ARGUMENTS
0000    ; AND EXPECTED FUNCTIONALITY PERFORMED.
0000    ;
0000    ; ENVIRONMENT:  USER MODE IMAGE; NEEDS CMKRNL PRIVILEGE,
0000    ;               DYNAMICALLY ACQUIRES OTHER PRIVILEGES, AS NEEDED.
0000    ;
0000    ; AUTHOR: THOMAS L. CAFARELLA,          CREATION DATE: MAY, 1978
0000    ;
0000    ; MODIFIED BY:
0000    ;
0000    ;          , : VERSION
0000    ; 01     -
0000    ;--
```

```
                0000          .SBTTL  DECLARATIONS
                0000    ;
                0000    ; INCLUDE FILES:
                0000    ;
                0000          $PRVDEF                         ; PRIVILEGE BIT DEFINITIONS
                0000          $PHDDEF                         ; PROCESS HEADER OFFSETS
                0000          $PSLDEF                         ; ACCESS MODE DEFINITIONS
                0000    ;
                0000    ; MACROS:
                0000    ;
                0000    ;
                0000    ; EQUATED SYMBOLS:
                0000    ;
                0000    ;
                0000    ;       THE FOLLOWING ERROR CODES MUST AGREE WITH THE CASE STATEMENT
                0000    ;       AT THE END OF THE VERIFY SUBROUTINE.
                0000    ;
00000001        0000    SCODE_ERR       = 1                     ; $ADJSTK STATUS CODE ERROR
00000002        0000    NEWADR_ERR      = 2                     ; NEWADR ARGUMENT ERROR
00000003        0000    ADJUSP_ERR      = 3                     ; ADJUSTED USER SP ERROR
00000004        0000    SCODE2_ERR      = 4                     ; 2ND $ADJSTK STATUS CODE ERROR
00000005        0000    RESTORE_ERR     = 5                     ; ERROR IN RESTORING ORIGINAL USP VALUE
                0000    ;
                0000    ; OWN STORAGE:
                0000    ;
```

```
00000000          .PSECT  RODATA,RD,NOWRT,NOEXE,LONG
    0000  TEST_MOD_NAME:: STRING  C,<SATSSS91>     ; TEST MODULE NAME
    0009  TEST_MOD_NAME_D: STRING I,<SATSSS91>     ; TEST MODULE NAME DESCRIPTOR
    0019  MSG1_INP_CTL:   STRING  I,<  SSADS!4ZW: CONDITIONS:>
    0039                                           ; FAO CTL STRING FOR MSG1 IN SUCCOMMON.MAR
    0039  MSG3_ERR_CTL::  STRING  I,< *SSADS!4ZW: !AS>
    0051                                           ; FAO CTL STRING FOR MSG3 IN SUCCOMMON.MAR
```

```
        00000000    88          .PSECT  RWDATA,RD,WRT,NOEXE,LONG
00000008  0000      89 PRIVMASK:        .BLKQ   1               ; ADDR OF PRIVILEGE MASK (IN PHD)
0000000C  0008      90 NEWADROUT:       .BLKL   1               ; OUTPUT FIELD FOR $ADJSTK NEWADR ARGUMENT
00000010  000C      91 ORIGUSP:         .BLKL   1               ; ORIGINAL VALUE OF USP (USER STACK POINTER)
00000014  0010      92 FINALUSP:        .BLKL   1               ; FINAL VALUE OF USP (USER STACK POINTER)
00000018  0014      93 SUM:             .BLKL   1               ; AREA FOR SUM OF ORIG NEWADR + ADJUST ARG
0000001C  0018      94 ADJUSP:          .BLKL   1               ; SAVE AREA FOR ADJUSTED USP VALUE
0000001D  001C      95 FAIL_CODE:       .BLKB   1               ; FAILURE CODE (SEE EQUATED SYMBOLS ABOVE)
```

```
                 001D    97          .SBTTL  CONDITION TABLES
                 001D    98 ;
                 001D    99 ;        ***** CONDITION TABLES FOR ADJSTK SYSTEM SERVICE *****
                 001D   100 ;
                 001D   101          COND    1,WORD,<ADJUST>,-
                 001D   102                    <ZERO VALUE>,-
                 001D   103                    <POSITIVE VALUE>,-
                 001D   104                    <NEGATIVE VALUE>,-
                 001D   105
            0000 005A   106                      .WORD       0
            0001 005C   107                      .WORD       1
            FFFC 005E   108                      .WORD       -4
                 0060   109 ;
                 0060   110          COND    2,LONG,<NEWADR>,-
                 0060   111                    <ZERO VALUE>,-
                 0060   112                    <NON-ZERO VALUE>,-
                 0060   113
        00000000 008A   114                      .LONG       0
        0000008E' 008E  115                      .ADDRESS    .
                 0092   116 ;
                 0092   117          COND    3,NULL
                 0093   118
                 0093   119          COND    4,NULL
                 0094   120
                 0094   121          COND    5,NULL
                 0095   122
            00000000    123          .PSECT  SATSSS91,RD,WRT,EXE
```

```
                                  0000          .SBTTL  TM_SETUP, TM_CLEANUP
                                  0000  ;++
                                  0000  ; FUNCTIONAL DESCRIPTION:
                                  0000  ;
                                  0000  ;                 TM_SETUP AND TM_CLEANUP ARE CALLED TO PERFORM
                                  0000  ; REQUIRED HOUSEKEEPING AT THE BEGINNING AND END, RESPECTIVELY, OF
                                  0000  ; TEST MODULE EXECUTION.
                                  0000  ;
                                  0000  ; CALLING SEQUENCE:
                                  0000  ;
                                  0000  ;       BSBW TM_SETUP   BSBW TM_CLEANUP
                                  0000  ;
                                  0000  ; INPUT PARAMETERS:
                                  0000  ;
                                  0000  ;       NONE
                                  0000  ;
                                  0000  ; IMPLICIT INPUTS:
                                  0000  ;
                                  0000  ;       NONE
                                  0000  ;
                                  0000  ; OUTPUT PARAMETERS:
                                  0000  ;
                                  0000  ;       NONE
                                  0000  ;
                                  0000  ; IMPLICIT OUTPUTS:
                                  0000  ;
                                  0000  ;       TM_SETUP:  COND TABLE INDEX REGISTERS (R2,3,4,5,6) CLEARED;
                                  0000  ;                  ALL PRIVILEGES ACQUIRED.
                                  0000  ;
                                  0000  ; COMPLETION CODES:
                                  0000  ;
                                  0000  ;       EFLAG SET TO NON-ZERO IF ERROR ENCOUNTERED.
                                  0000  ;
                                  0000  ; SIDE EFFECTS:
                                  0000  ;
                                  0000  ;       SS_CHECK AND ERR_EXIT MACROS CAUSE PREMATURE EXIT
                                  0000  ;       (VIA RSB) IF ERROR ENCOUNTERED.
                                  0000  ;
                                  0000  ;--
                                  0000
                                  0000
                                  0000
                                  0000  TM_SETUP::
                        52    D4  0000          CLRL    R2                      ; INITIALIZE
                        53    D4  0002          CLRL    R3                      ; .. CONDITION
                        54    D4  0004          CLRL    R4                      ; .... TABLE
                        55    D4  0006          CLRL    R5                      ; ...... INDEX
                        56    D4  0008          CLRL    R6                      ; ........ REGISTERS
                      FFF3'   30  000A          BSBW    MOD_MSG_PRINT           ; PRINT TEST MODULE BEGIN MSG
00000000'EF   00000000'EF   DE  000D          MOVAL   TEST_MOD_SUCC,TMD_ADDR  ; ASSUME END MSG WILL SHOW SUCCESS
   03   00   00000000'8F   F0  0018          INSV    #SUCCESS,#0,#3,MOD_MSG_CODE ; ADJUST STATUS CODE FOR SUCCESS
             00000000'EF      0020
                              0025          MODE    TO,5$,KRNL              ; KERNEL MODE TO ACCESS PHD
        59   00000000'9F   D0  0048          MOVL    @#CTL$GL_PHD,R9         ; GET PROCESS HEADER ADDRESS
        00000000'EF   69   DE  004F          MOVAL   PHD$Q_PRIVMSK(R9),PRIVMASK ; GET PRIV MASK ADDRESS
                              0056          MODE    FROM,5$ ; BACK TO USER MODE
                              0057          PRIV    ADD,ALL                 ; GET ALL PRIVILEGES
```

```
                0077   181          $SETPRN_S TEST_MOD_NAME_D        ; SET PROCESS NAME
                0084   182          SS_CHECK  NORMAL                 ; CHECK STATUS CODE RETURNED FROM SETPRN
          05    00B2   183          RSB                              ; RETURN TO MAIN ROUTINE
                00B3   184 TM_CLEANUP::
   FF4A'  30    00B3   185          BSBW    MOD_MSG_PRINT            ; PRINT TEST MODULE END MSG
          05    00B6   186          RSB                              ; RETURN TO MAIN ROUTINE
```

```
00B7          .SBTTL  CONDITION SUBROUTINES - SETUP AND CLEANUP
00B7  ;++
00B7  ; FUNCTIONAL DESCRIPTION:
00B7  ;
00B7  ;                 CONDX AND CONDX_CLEANUP ARE SUBROUTINES WHICH ARE EXECUTED
00B7  ; BEFORE AND AFTER THE VERIFY SUBROUTINE, RESPECTIVELY, WHENEVER A NEW
00B7  ; CONDITION X VALUE IS SELECTED (SEE FUNCTIONAL DESCRIPTION OF SUCCOMMON
00B7  ; ROUTINE IN SUCCOMMON.MAR). ANY SETUP FUNCTION PARTICULAR TO THE
00B7  ; CONDITION X TABLE IS INCLUDED IN THE CONDX SUBROUTINE AND CLEANED
00B7  ; UP, IF NECESSARY, IN THE CONDX_CLEANUP SUBROUTINE. THIS INCLUDES,
00B7  ; ESPECIALLY, CODE TO DETECT CONFLICTS AMONG CURRENT ENTRIES IN TWO
00B7  ; OR MORE CONDITION TABLES. IF A CONFLICT IS DETECTED, A NON-ZERO
00B7  ; VALUE IS STORED INTO CONFLICT, WHICH CAUSES THE CALLING ROUTINE
00B7  ; (SUCCOMMON) TO SKIP THE CURRENT ENTRY IN THE CONDITION X TABLE.
00B7  ;
00B7  ; CALLING SEQUENCE:
00B7  ;
00B7  ;       BSBW CONDX   BSBW CONDX_CLEANUP
00B7  ;         WHERE X = 1,2,3,4,5
00B7  ;
00B7  ; INPUT PARAMETERS:
00B7  ;
00B7  ;       CONFLICT = 0
00B7  ;
00B7  ; IMPLICIT INPUTS:
00B7  ;
00B7  ;       R2,3,4,5,6 CONTAIN CURRENT CONDITION TABLE INDEX VALUES
00B7  ;         FOR COND TABLES 1,2,3,4,5, RESPECTIVELY.
00B7  ;
00B7  ; OUTPUT PARAMETERS:
00B7  ;
00B7  ;       CONFLICT SET TO NON-ZERO IF COND TABLE CONFLICT DETECTED.
00B7  ;
00B7  ; IMPLICIT OUTPUTS:
00B7  ;
00B7  ;       R2,3,4,5,6 PRESERVED
00B7  ;
00B7  ; COMPLETION CODES:
00B7  ;
00B7  ;       NONE
00B7  ;
00B7  ; SIDE EFFECTS:
00B7  ;
00B7  ;       NONE
00B7  ;
00B7  ;--
00B7
00B7
00B7
00B7  COND1::
05 00B7               RSB                                     ; RETURN TO MAIN ROUTINE
00B8  COND1_CLEANUP::
05 00B8               RSB                                     ; RETURN TO MAIN ROUTINE
00B9  COND2::
05 00B9               RSB                                     ; RETURN TO MAIN ROUTINE
00BA  COND2_CLEANUP::
05 00BA               RSB                                     ; RETURN TO MAIN ROUTINE
```

```
        00BB   245 COND3::
 05     00BB   246         RSB                                  ; RETURN TO MAIN ROUTINE
        00BC   247 COND3_CLEANUP::
 05     00BC   248         RSB                                  ; RETURN TO MAIN ROUTINE
        00BD   249 COND4::
 05     00BD   250         RSB                                  ; RETURN TO MAIN ROUTINE
        00BE   251 COND4_CLEANUP::
 05     00BE   252         RSB                                  ; RETURN TO MAIN ROUTINE
        00BF   253 COND5::
 05     00BF   254         RSB                                  ; RETURN TO MAIN ROUTINE
        00C0   255 COND5_CLEANUP::
 05     00C0   256         RSB                                  ; RETURN TO MAIN ROUTINE
```

```
                                00C1          .SBTTL  FORM_CONDS
                                00C1  ;++
                                00C1  ; FUNCTIONAL DESCRIPTION:
                                00C1  ;
                                00C1  ;                 FORM_CONDS FORMATS AND PRINTS INFORMATION ABOUT
                                00C1  ;  THE CURRENT ELEMENT IN EACH OF THE CONDITION TABLES.
                                00C1  ;
                                00C1  ; CALLING SEQUENCE:
                                00C1  ;
                                00C1  ;       BSBW FORM_CONDS
                                00C1  ;
                                00C1  ; INPUT PARAMETERS:
                                00C1  ;
                                00C1  ;       NONE
                                00C1  ;
                                00C1  ; IMPLICIT INPUTS:
                                00C1  ;
                                00C1  ;       R2,3,4,5,6 CONTAIN CURRENT CONDITION TABLE INDEX VALUES
                                00C1  ;         FOR COND TABLES 1,2,3,4,5, RESPECTIVELY.
                                00C1  ;       FOR X = 1,2,3,4,5 :
                                00C1  ;               CONDX_T - TITLE TEXT FOR CONDX TABLE
                                00C1  ;               CONDX_TAB - ELEMENT TEXT FOR CONDX TABLE
                                00C1  ;               CONDX_C - CONTEXT OF THE CONDX TABLE
                                00C1  ;               CONDX_E - DATA ELEMENTS OF THE CONDX TABLE
                                00C1  ;
                                00C1  ; OUTPUT PARAMETERS:
                                00C1  ;
                                00C1  ;       NONE
                                00C1  ;
                                00C1  ; IMPLICIT OUTPUTS:
                                00C1  ;
                                00C1  ;       NONE
                                00C1  ;
                                00C1  ; COMPLETION CODES:
                                00C1  ;
                                00C1  ;       NONE
                                00C1  ;
                                00C1  ; SIDE EFFECTS:
                                00C1  ;
                                00C1  ;       NONE
                                00C1  ;
                                00C1  ;--
                                00C1
                                00C1
                                00C1  FORM_CONDS::
                                00C1          $FAO_S  MSG1_INP_CTL,FAO_LEN,FAO_DESC,TESTNUM
                                00E0                                          ; FORMAT CONDITIONS HEADER MSG
                     FF1D'  30  00E0          BSBW    OUTPUT_MSG             ; ... AND PRINT IT
                 14    02  91  00E3          CMPB    #COND1_C,#NULL         ; IS CONDITION 1 NULL ?
                       03  12  00E6          BNEQU   10$                    ; NO -- CONTINUE
                     00D7  31  00E8          BRW     FORM_CONDSX            ; YES -- SUBROUTINE IS FINISHED
                               00EB  10$:
  00000000'EF   0000001D'EF  DE  00EB          MOVAL   COND1_T,MSG_A          ; SAVE ADDRESS OF CONDITION 1 TITLE FOR FAO
00000000'EF   00000025'EF42  D0  00F6          MOVL    COND1_TAB[R2],MSG_B    ; SAVE ADDR OF COND 1 CURR TEXT ELT FOR FAO
          00000000'EF    02  90  0102          MOVB    #COND1_C,MSG_CTXT      ; SAVE CONDITION 1 CONTEXT FOR FAO
                               0109          MOV_VAL COND1_E[R2],MSG_DATA1 ; GIVE COND 1 DATA VALUE TO FAO
```

```
                    FEE8'  30  0115   315          BSBW    WRITE_MSG2                  ; FORMAT AND WRITE CONDITION 1 MSG
                 14   04   91  0118   316          CMPB    #COND2_C,#NULL              ; IS CONDITION 2 NULL ?
                      03   12  011B   317          BNEQU   20$                         ; NO -- CONTINUE
                    00A2   31  011D   318          BRW     FORM_CONDSX                 ; YES -- SUBROUTINE IS FINISHED
                           0120   319 20$:
  00000000'EF   00000060'EF  DE  0120   320          MOVAL   COND2_T,MSG_A               ; SAVE ADDRESS OF CONDITION 2 TITLE FOR FAO
00000000'EF   00000068'EF43  D0  012B   321          MOVL    COND2_TAB[R3],MSG_B         ; SAVE ADDR OF COND 2 CURR TEXT ELT FOR FAO
         00000000'EF   04   90  0137   322          MOVB    #COND2_C,MSG_CTXT           ; SAVE CONDITION 2 CONTEXT FOR FAO
                           013E   323          MOV_VAL COND2_C,COND2_E[R3],MSG_DATA1 ; GIVE COND 2 DATA VALUE TO FAO
                    FEB3'  30  014A   324          BSBW    WRITE_MSG2                  ; FORMAT AND WRITE CONDITION 2 MSG
                 14   14   91  014D   325          CMPB    #COND3_C,#NULL              ; IS CONDITION 3 NULL ?
                      03   12  0150   326          BNEQU   30$                         ; NO -- CONTINUE
                    006D   31  0152   327          BRW     FORM_CONDSX                 ; YES -- SUBROUTINE IS FINISHED
                           0155   328 30$:
  00000000'EF   00000092'EF  DE  0155   329          MOVAL   COND3_T,MSG_A               ; SAVE ADDRESS OF CONDITION 3 TITLE FOR FAO
00000000'EF   00000092'EF44  D0  0160   330          MOVL    COND3_TAB[R4],MSG_B         ; SAVE ADDR OF COND 3 CURR TEXT ELT FOR FAO
         00000000'EF   14   90  016C   331          MOVB    #COND3_C,MSG_CTXT           ; SAVE CONDITION 3 CONTEXT FOR FAO
                           0173   332          MOV_VAL COND3_C,COND3_E[R4],MSG_DATA1 ; GIVE COND 3 DATA VALUE TO FAO
                    FE8A'  30  0173   333          BSBW    WRITE_MSG2                  ; FORMAT AND WRITE CONDITION 3 MSG
                 14   14   91  0176   334          CMPB    #COND4_C,#NULL              ; IS CONDITION 4 NULL ?
                      47   13  0179   335          BEQLU   FORM_CONDSX                 ; YES -- SUBROUTINE IS FINISHED
  00000000'EF   00000093'EF  DE  017B   336          MOVAL   COND4_T,MSG_A               ; SAVE ADDRESS OF CONDITION 4 TITLE FOR FAO
00000000'EF   00000093'EF45  D0  0186   337          MOVL    COND4_TAB[R5],MSG_B         ; SAVE ADDR OF COND 4 CURR TEXT ELT FOR FAO
         00000000'EF   14   90  0192   338          MOVB    #COND4_C,MSG_CTXT           ; SAVE CONDITION 4 CONTEXT FOR FAO
                           0199   339          MOV_VAL COND4_C,COND4_E[R5],MSG_DATA1 ; GIVE COND 4 DATA VALUE TO FAO
                    FE64'  30  0199   340          BSBW    WRITE_MSG2                  ; FORMAT AND WRITE CONDITION 4 MSG
                 14   14   91  019C   341          CMPB    #COND5_C,#NULL              ; IS CONDITION 5 NULL ?
                      21   13  019F   342          BEQLU   FORM_CONDSX                 ; YES -- SUBROUTINE IS FINISHED
  00000000'EF   00000094'EF  DE  01A1   343          MOVAL   COND5_T,MSG_A               ; SAVE ADDRESS OF CONDITION 5 TITLE FOR FAO
00000000'EF   00000094'EF46  D0  01AC   344          MOVL    COND5_TAB[R6],MSG_B         ; SAVE ADDR OF COND 5 CURR TEXT ELT FOR FAO
         00000000'EF   14   90  01B8   345          MOVB    #COND5_C,MSG_CTXT           ; SAVE CONDITION 5 CONTEXT FOR FAO
                           01BF   346          MOV_VAL COND5_C,COND5_E[R6],MSG_DATA1 ; GIVE COND 5 DATA VALUE TO FAO
                    FE3E'  30  01BF   347          BSBW    WRITE_MSG2                  ; FORMAT AND WRITE CONDITION 5 MSG
                           01C2   348 FORM_CONDSX:
                        05 01C2   349          RSB                                 ; RETURN TO CALLER
```

```
01C3          .SBTTL  VERIFY
01C3    ;++
01C3    ; FUNCTIONAL DESCRIPTION:
01C3    ;
01C3    ;                 VERIFY IS CALLED ONCE FOR EACH COMBINATION OF CONDITION
01C3    ; TABLE VALUES (AS DETERMINED BY THE INDEX REGISTERS R2,3,4,5,6 FOR
01C3    ; COND TABLES 1,2,3,4,5, RESPECTIVELY). VERIFY ESTABLISHES THE CONDITIONS
01C3    ; SPECIFIED BY THE COND TABLES AND ISSUES THE SUBJECT SYSTEM SERVICE
01C3    ; ($ADJSTK). THEN, THE SUCCESSFUL OPERATION OF THE SERVICE IS VERIFIED
01C3    ; BY EXAMINING THE STATUS CODE RETURNED, THE VALUES FOR RETURN ARGUMENTS
01C3    ; AND THE FUNCTIONALITY PERFORMED. THE EXAMINATIONS TAKE THE FORM OF
01C3    ; COMPARISONS AGAINST EXPECTED VALUES. ANY FAILING COMPARISON CAUSES AN
01C3    ; ERR_EXIT MACRO TO BE EXECUTED (EITHER DIRECTLY, OR INDIRECTLY,
01C3    ; THROUGH THE SS_CHECK MACRO); ERR_EXIT SETS EFLAG TO NON-ZERO,
01C3    ; PRINTS ERROR MESSAGES AND CAUSES AN IMMEDIATE RSB TO CALLER.
01C3    ; WHEN ERR_EXIT IS EXECUTED, FURTHER CALLS TO VERIFY ARE SUPPRESSED,
01C3    ; AND, AFTER EXECUTING CLEANUP SUBROUTINES, THE IMAGE EXITS.
01C3    ;
01C3    ; CALLING SEQUENCE:
01C3    ;
01C3    ;       BSBW VERIFY
01C3    ;
01C3    ; INPUT PARAMETERS:
01C3    ;
01C3    ;       NONE
01C3    ;
01C3    ; IMPLICIT INPUTS:
01C3    ;
01C3    ;       R2,3,4,5,6 CONTAIN CURRENT CONDITION TABLE INDEX VALUES
01C3    ;         FOR COND TABLES 1,2,3,4,5, RESPECTIVELY.
01C3    ;       FOR X = 1,2,3,4,5 :
01C3    ;               CONDX_E - ADDRESS OF TABLE OF DATA VALUES FOR CONDX
01C3    ;                 TABLE. IF THE CONTEXT OF TABLE X IS A SYSTEM SERVICE
01C3    ;                 ARGUMENT, THE ARGUMENT NAME MAY BE USED AS A SYNONYM
01C3    ;                 FOR CONDX_E.
01C3    ;
01C3    ; OUTPUT PARAMETERS:
01C3    ;
01C3    ;       NONE
01C3    ;
01C3    ; IMPLICIT OUTPUTS:
01C3    ;
01C3    ;       VERIFY HAS NO OUTPUT. SINCE ITS PURPOSE IS TO TEST FOR ERRORS,
01C3    ;       IT MERELY RETURNS TO CALLER NORMALLY AFTER THE TESTS, PROVIDING
01C3    ;       ALL WERE SUCCESSFUL; IF AN ERROR IS DISCOVERED, RETURN IS VIA
01C3    ;       AN ERR_EXIT OR SS_CHECK MACRO, BOTH OF WHICH DOCUMENT DETECTED
01C3    ;       ERRORS.
01C3    ;
01C3    ; COMPLETION CODES:
01C3    ;
01C3    ;       EFLAG SET TO NON-ZERO IF ERROR ENCOUNTERED.
01C3    ;
01C3    ; SIDE EFFECTS:
01C3    ;
01C3    ;       SS_CHECK AND ERR_EXIT MACROS CAUSE PREMATURE EXIT
01C3    ;       (VIA RSB) IF ERROR ENCOUNTERED.
01C3    ;
```

```
                                01C3   408 ;--
                                01C3   409
                                01C3   410
                                01C3   411
                                01C3   412 VERIFY::
          00000000'EF   95      01C3   413          TSTB    CFLAG                   ; SHOULD CONDITIONS BE PRINTED ?
                   03   13      01C9   414          BEQL    5$                      ; NO -- CONTINUE
                 FEF3   30      01CB   415          BSBW    FORM_CONDS              ; YES -- FMT & PRINT ALL CONDS FOR THIS T.C.
                                01CE   416 5$:
          0000001C'EF   94      01CE   417          CLRB    FAIL_CODE               ; INDICATE NO FAILURE YET FOR THIS TEST CASE
     57   0000005A'EF42   32    01D4   418          CVTWL   ADJUST[R2],R7           ; SIGN-EXTEND ADJUST VALUE TO LONGWORD
     0000008A'EF43   57   C1    01DC   419          ADDL3   R7,NEWADR[R3],SUM       ; GET VALUE FOR USP (USER STACK PTR)
          00000014'EF           01E4
                                01E9   420          MODE    TO,10$,KRNL             ; INTO KERNEL MODE TO AVOID USER SP
 0000000C'EF   00000000'8F   DB 020C   421          MFPR    #PR$_USP,ORIGUSP        ; REMEMBER USER SP VALUE BEFORE TEST
00000008'EF   0000008A'EF43   D0 0217   422          MOVL    NEWADR[R3],NEWADROUT    ; SEPARATE LONGWORD FOR $ADJSTK OUTPUT
                                0223   423 ;
                                0223   424 ; ****** SYSTEM SERVICE CALL WHICH IS THE SUBJECT OF THIS TEST CASE ******
                                0223   425 ;
                                0223   426          $ADJSTK_S ACMODE=#PSL$C_USER, -
                                0223   427                    ADJUST=ADJUST[R2],  -
                                0223   428                    NEWADR=NEWADROUT
     00000000'8F   50   D1      023A   429          CMPL    R0,#SS$_NORMAL          ; CODE RECEIVED = CODE EXPECTED ?
                   19   13      0241   430          BEQLU   20$                     ; YES -- GO DO SOME VERIFYING
 00000000'EF   00000000'8F   D0 0243   431          MOVL    #SS$_NORMAL,EXPV        ; NO -- LOAD UP EXPECTED AND
     00000000'EF   50   D0      024E   432          MOVL    R0,RECV                 ; ... RECEIVED VALUES
     0000001C'EF   01   90      0255   433          MOVB    #SCODE_ERR,FAIL_CODE    ; ... AND INDICATE ERROR FOR LATER
                                025C   434 20$:
 00000018'EF   00000000'8F   DB 025C   435          MFPR    #PR$_USP,ADJUSP         ; SAVE "ADJUSTED" USP VALUE
          0000001C'EF   95      0267   436          TSTB    FAIL_CODE               ; ANY ERRORS ENCOUNTERED YET ?
                   2A   12      026D   437          BNEQ    30$                     ; YES -- SKIP THIS VERIFICATION
 00000018'EF   00000008'EF   D1 026F   438          CMPL    NEWADROUT,ADJUSP        ; DID NEWADR GET "ADJUSTED" VALUE ?
                   1D   13      027A   439          BEQLU   30$                     ; YES -- THAT'S OK
 00000000'EF   00000018'EF   D0 027C   440          MOVL    ADJUSP,EXPV             ; NO -- LOAD UP EXPECTED AND
 00000000'EF   00000008'EF   D0 0287   441          MOVL    NEWADROUT,RECV          ; ... RECEIVED VALUES
     0000001C'EF   02   90      0292   442          MOVB    #NEWADR_ERR,FAIL_CODE   ; ... AND INDICATE ERROR FOR LATER
                                0299   443 30$:
          0000001C'EF   95      0299   444          TSTB    FAIL_CODE               ; ANY ERRORS ENCOUNTERED YET ?
                   39   12      029F   445          BNEQ    50$                     ; YES -- SKIP THIS VERIFICATION
     57   00000014'EF   D0      02A1   446          MOVL    SUM,R7                  ; SUM OF ORIGINAL NEWADR + ADJUST VALUE
     0000008A'EF43   D5         02A8   447          TSTL    NEWADR[R3]              ; WAS NEWADR ORIGINALLY ZERO ?
                   07   12      02AF   448          BNEQ    40$                     ; NO -- LEAVE R7 AS IS
     57   0000000C'EF   C0      02B1   449          ADDL2   ORIGUSP,R7              ; YES -- ADD ORIGUSP TO ADJUST VALUE
                                02B8   450 40$:
     00000018'EF   57   D1      02B8   451          CMPL    R7,ADJUSP               ; IS ADJUSP WHAT WE EXPECTED ?
                   19   13      02BF   452          BEQLU   50$                     ; YES -- GO ON TO NEXT VERIFICATION
     00000000'EF   57   D0      02C1   453          MOVL    R7,EXPV                 ; NO -- LOAD UP EXPECTED AND
 00000000'EF   00000018'EF   D0 02C8   454          MOVL    ADJUSP,RECV             ; ... RECEIVED VALUES
     0000001C'EF   03   90      02D3   455          MOVB    #ADJUSP_ERR,FAIL_CODE   ; ... AND INDICATE ERROR FOR LATER
                                02DA   456 50$:
 00000010'EF   0000000C'EF   D0 02DA   457          MOVL    ORIGUSP,FINALUSP        ; SET ORIGINAL USP VALUE UP FOR FINAL
                                02E5   458          $ADJSTK_S ACMODE=#PSL$C_USER, - ; RE-ADJUST USER STACK BACK TO ORIGINAL
                                02E5   459                    NEWADR=FINALUSP
          0000001C'EF   95      02F7   460          TSTB    FAIL_CODE               ; ANY ERRORS ENCOUNTERED YET ?
                   22   12      02FD   461          BNEQ    60$                     ; YES -- SKIP THIS VERIFICATION
     00000000'8F   50   D1      02FF   462          CMPL    R0,#SS$_NORMAL          ; CODE RECEIVED = CODE EXPECTED ?
                   19   13      0306   463          BEQLU   60$                     ; YES -- ON TO MORE VERIFICATION
```

```
00000000'EF   00000000'8F   D0  0308   MOVL    #SS$_NORMAL,EXPV          ; NO -- LOAD UP EXPECTED AND
        00000000'EF   50    D0  0313   MOVL    R0,RECV                   ; ... RECEIVED VALUES
        0000001C'EF   04    90  031A   MOVB    #SCODE2_ERR,FAIL_CODE     ; ... AND INDICATE ERROR FOR LATER
                            0321   60$:
        57   00000000'8F    DB  0321   MFPR    #PR$_USP,R7               ; GET CURRENT USP VALUE ...
        0000000C'EF   57    D1  0328   CMPL    R7,ORIGUSP                ; ... WAS IT RESTORED TO ORIGINAL ?
                      2C    13  032F   BEQLU   70$                       ; YES -- ALL FINISHED WITH VERIFICATION
0000000C'EF   00000000'EF   DA  0331   MTPR    PR$_USP,ORIGUSP           ; NO -- RESTORE ORIGINAL USP VALUE
              0000001C'EF   95  033C   TSTB    FAIL_CODE                 ; ANY ERRORS ENCOUNTERED YET ?
                      19    12  0342   BNEQ    70$                       ; YES -- DON'T REPORT THIS ONE
00000000'EF   0000000C'EF   D0  0344   MOVL    ORIGUSP,EXPV              ; NO -- LOAD UP EXPECTED AND
        00000000'EF   57    D0  034F   MOVL    R7,RECV                   ; ... RECEIVED VALUES
        0000001C'EF   05    90  0356   MOVB    #RESTORE_ERR,FAIL_CODE    ; ... AND INDICATE ERROR FOR LATER
                            035D   70$:
                            035D       MODE    FROM,10$                  ; GET BACK TO USER MODE
                            035E   ;
                            035E   ; THE FOLLOWING CASE STATEMENT CHOOSES AN APPROPRIATE ERROR MESSAGE
                            035E   ; BASED ON A VALUE IN FAIL_CODE. A VALUE OF ZERO IMPLIES NO ERRORS.
                            035E   ; EACH ERR_EXIT MACRO TERMINATES THE VERIFY SUBROUTINE. THE ERR_EXIT
                            035E   ; MACROS APPEAR HERE BECAUSE THEY MAY NOT BE ISSUED IN A KERNEL MODE
                            035E   ; ROUTINE.
                            035E   ;
   05   00   0000001C'EF    8F  035E   CASEB   FAIL_CODE,#0,#5           ; BRANCH TO CORRECT ERR_EXIT MACRO
                            0366   80$:                                  ; START OF CASE WORD DISPLACEMENTS
                     01BD'  0366       .WORD   VERIFYX-80$               ; FAIL_CODE = 0, NO ERRORS
                     000F'  0368       .WORD   90$-80$                   ; FAIL_CODE = SCODE_ERR
                     005E'  036A       .WORD   100$-80$                  ; FAIL_CODE = NEWADR_ERR
                     00BF'  036C       .WORD   110$-80$                  ; FAIL_CODE = ADJUSP_ERR
                     010C'  036E       .WORD   120$-80$                  ; FAIL_CODE = SCODE2_ERR
                     0160'  0370       .WORD   130$-80$                  ; FAIL_CODE = RESTORE_ERR
                 01AE   31  0372       BRW     VERIFYX                   ; BRANCH PAST CASE ROUTINES
                            0375   90$:
                            0375       ERR_EXIT LONG,<INCORRECT STATUS CODE RETURNED FROM ADJSTK>
                            03C4   100$:
                            03C4       ERR_EXIT LONG,<ADJUSTED STACK POINTER VALUE NOT>, -
                            03C4                     < RETURNED IN NEWADR ARGUMENT>
                            0425   110$:
                            0425       ERR_EXIT LONG,<UNEXPECTED RESULTANT STACK POINTER VALUE>
                            0472   120$:
                            0472       ERR_EXIT LONG,<UNEXPECTED STATUS CODE FROM NON-SUBJECT $ADJSTK>
                            04C6   130$:
                            04C6       ERR_EXIT LONG,<2ND $ADJSTK DID NOT RE-ADJUST USP BACK TO>, -
                            04C6                     < ORIGINAL VALUE>
                            0523   ;
                            0523   ; END OF CASE ROUTINES
                            0523   ;
                            0523   VERIFYX:
                        05  0523       RSB                               ; RETURN TO CALLER
```

```
0524          .SBTTL  VFY_CLEANUP
0524  ;++
0524  ; FUNCTIONAL DESCRIPTION:
0524  ;
0524  ;               VFY_CLEANUP EXECUTES SYSTEM SERVICES TO UNDO THE
0524  ; EFFECT OF THOSE ISSUED IN THE VERIFY SUBROUTINE. VFY_CLEANUP MUST
0524  ; ASSUME THAT VERIFY MAY NOT HAVE EXECUTED IN ITS ENTIRETY (IF AN
0524  ; ERROR IS FOUND). ALSO, VFY_CLEANUP MAY ISSUE SS_CHECK OR ERR_EXIT
0524  ; ONLY AFTER PERFORMING ALL OF ITS CLEANUP OPERATIONS; THIS IS REQUIRED
0524  ; IN THE EVENT THAT VFY_CLEANUP IS CALLED DURING ERROR PROCESSING,
0524  ; WHEN PERFORMING THE REQUIRED CLEANUP IS MORE IMPORTANT THAN
0524  ; POSSIBLY DISCOVERING A SECOND ERROR.
0524  ;
0524  ; CALLING SEQUENCE:
0524  ;
0524  ;       BSBW VFY_CLEANUP
0524  ;
0524  ; INPUT PARAMETERS:
0524  ;
0524  ;       NONE
0524  ;
0524  ; IMPLICIT INPUTS:
0524  ;
0524  ;       R2,3,4,5,6 CONTAIN CURRENT CONDITION TABLE INDEX VALUES
0524  ;         FOR COND TABLES 1,2,3,4,5, RESPECTIVELY.
0524  ;       FOR X = 1,2,3,4,5 :
0524  ;               CONDX_E - ADDRESS OF TABLE OF DATA VALUES FOR CONDX
0524  ;                 TABLE. IF THE CONTEXT OF TABLE X IS A SYSTEM SERVICE
0524  ;                 ARGUMENT, THE ARGUMENT NAME MAY BE USED AS A SYNONYM
0524  ;                 FOR CONDX_E.
0524  ;
0524  ; OUTPUT PARAMETERS:
0524  ;
0524  ;       NONE
0524  ;
0524  ; IMPLICIT OUTPUTS:
0524  ;
0524  ;       NONE
0524  ;
0524  ; COMPLETION CODES:
0524  ;
0524  ;       EFLAG SET TO NON-ZERO IF ERROR ENCOUNTERED.
0524  ;
0524  ; SIDE EFFECTS:
0524  ;
0524  ;       SS_CHECK AND ERR_EXIT MACROS CAUSE PREMATURE EXIT
0524  ;       (VIA RSB) IF ERROR ENCOUNTERED.
0524  ;
0524  ;--
0524
0524
0524
0524  VFY_CLEANUP::
05 0524       RSB                                     ; RETURN TO CALLER
0525          .END
```

```
$$$$                    = 000004D0 R     04
$$$CHARS                = 00000038
$$$CHARS1               = 0000000A
$$$CHARS2               = 0000000E
$$$CHARS3               = 00000000
$$$CHARS4               = 00000000
$$$CHARS5               = 00000000
$$$COND_A               = 00000001
$$$STRINGS              = 00000001
$$$STRINGS2             = 00000005
$$T2                    = 00000004
ADJUSP                    00000018 R     03
ADJUSP_ERR              = 00000003
ADJUST                    0000005A R     03
BYTE                    = 00000001  G
CFLAG                     ********   X   04
CHMRTN                    ********   X   04
CHM_CONT                  ********   X   04
COMP_SC                   ********   X   04
COND1                     000000B7 RG    04
COND1_C                 = 00000002
COND1_CLEANUP             000000B8 RG    04
COND1_E                   0000005A R     03
COND1_H                   00000024 RG    03
COND1_T                   0000001D R     03
COND1_TAB                 00000025 R     03
COND2                     000000B9 RG    04
COND2_C                 = 00000004
COND2_CLEANUP             000000BA RG    04
COND2_E                   0000008A R     03
COND2_H                   00000067 RG    03
COND2_T                   00000060 R     03
COND2_TAB                 00000068 R     03
COND3                     000000BB RG    04
COND3_C                 = 00000014
COND3_CLEANUP             000000BC RG    04
COND3_H                   00000092 RG    03
COND3_T                   00000092 R     03
COND3_TAB                 00000092 R     03
COND4                     000000BD RG    04
COND4_C                 = 00000014
COND4_CLEANUP             000000BE RG    04
COND4_H                   00000093 RG    03
COND4_T                   00000093 R     03
COND4_TAB                 00000093 R     03
COND5                     000000BF RG    04
COND5_C                 = 00000014
COND5_CLEANUP             000000C0 RG    04
COND5_H                   00000094 RG    03
COND5_T                   00000094 R     03
COND5_TAB                 00000094 R     03
CTL$GL_PHD                ********   X   04
DESC                    = 00000010  G
EFLAG                     ********   X   04
EXPV                      ********   X   04
FAIL_CODE                 0000001C R     03
FAO_DESC                  ********   X   04
FAO_LEN                   ********   X   04
FINALUSP                  00000010 R     03
FORM_CONDS                000000C1 RG    04
FORM_CONDSX               000001C2 R     04
LONG                    = 00000004  G
MOD_MSG_CODE              ********   X   04
MOD_MSG_PRINT             ********   X   04
MSG1_INP_CTL              00000019 R     02
MSG3_ERR_CTL              00000039 RG    02
MSG_A                     ********   X   04
MSG_B                     ********   X   04
MSG_CTXT                  ********   X   04
MSG_DATA1                 ********   X   04
NEWADR                    0000008A R     03
NEWADROUT                 00000008 R     03
NEWADR_ERR              = 00000002
NOTARG                  = 00000000  G
NULL                    = 00000014  G
ORIGUSP                   0000000C R     03
OUTPUT_MSG                ********   X   04
PCV                       ********   X   04
PHD$Q_PRIVMSK           = 00000000
PR$_USP                   ********   X   04
PRIVMASK                  00000000 R     03
PRIV_ARGS               = 00000002
PROCESS_ERR               ********   X   04
PSL$C_USER              = 00000003
QUAD                    = 00000008  G
RECV                      ********   X   04
RESTORE_ERR             = 00000005
REST_REGS                 ********   X   04
SAVE_REGS                 ********   X   04
SCODE2_ERR              = 00000004
SCODE_ERR               = 00000001
SS$_NORMAL                ********   X   04
SUCCESS                   ********   X   04
SUM                       00000014 R     03
SYS$ADJSTK                ********  GX   04
SYS$CMKRNL                ********  GX   04
SYS$FAO                   ********   X   04
SYS$SETPRN                ********  GX   04
SYS$SETPRV                ********  GX   04
TESTNUM                   ********   X   04
TEST_MOD_NAME             00000000 RG    02
TEST_MOD_NAME_D           00000009 R     02
TEST_MOD_SUCC             ********   X   04
TMD_ADDR                  ********   X   04
TM_CLEANUP                000000B3 RG    04
TM_SETUP                  00000000 RG    04
VERIFY                    000001C3 RG    04
VERIFYX                   00000523 R     04
VFY_CLEANUP               00000524 RG    04
WORD                    = 00000002  G
WRITE_MSG2                ********   X   04
```

```
                                  +-----------------+
                                  ! Psect synopsis !
                                  +-----------------+

PSECT name                      Allocation          PSECT No.  Attributes
----------                      ----------          ---------  ----------
.  ABS  .                       00000000  (    0.)  00 (  0.)  NOPIC   USR   CON   ABS   LCL NOSHR NOEXE NORD  NOWRT NOVEC BYTE
$ABS$                           00000000  (    0.)  01 (  1.)  NOPIC   USR   CON   ABS   LCL NOSHR   EXE   RD    WRT NOVEC BYTE
RODATA                          00000051  (   81.)  02 (  2.)  NOPIC   USR   CON   REL   LCL NOSHR NOEXE   RD  NOWRT NOVEC LONG
RWDATA                          00000095  (  149.)  03 (  3.)  NOPIC   USR   CON   REL   LCL NOSHR NOEXE   RD    WRT NOVEC LONG
SATSSS91                        00000525  ( 1317.)  04 (  4.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD    WRT NOVEC BYTE

                                +-------------------------+
                                ! Performance indicators !
                                +-------------------------+

Phase                     Page faults    CPU Time       Elapsed Time
-----                     -----------    --------       ------------
Initialization                     35    00:00:00.09    00:00:00.28
Command processing                132    00:00:00.62    00:00:02.10
Pass 1                            244    00:00:06.52    00:00:10.93
Symbol table sort                   0    00:00:00.50    00:00:00.52
Pass 2                            116    00:00:01.78    00:00:02.88
Symbol table output                13    00:00:00.09    00:00:00.09
Psect synopsis output               3    00:00:00.03    00:00:00.03
Cross-reference output              0    00:00:00.00    00:00:00.00
Assembler run totals              545    00:00:09.63    00:00:16.83

The working set limit was 1500 pages.
34049 bytes (67 pages) of virtual memory were used to buffer the intermediate code.
There were 20 pages of symbol table space allocated to hold 343 non-local and 32 local symbols.
567 source lines were read in Pass 1, producing 23 object records in Pass 2.
33 pages of virtual memory were used to define 24 macros.

                                  +---------------------------+
                                  ! Macro library statistics !
                                  +---------------------------+

Macro library name                            Macros defined
------------------                            --------------
_$255$DUA28:[SHRLIB]UETP.MLB;1                             9
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                             1
_$255$DUA28:[SYSLIB]STARLET.MLB;2                         11
TOTALS (all libraries)                                    21

636 GETS were required to define 21 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:SATSSS91/OBJ=OBJ$:SATSSS91 MSRC$:SATSSS91/UPDATE=(ENH$:SATSSS91)+EXECML$/LIB+SHRLIB$:UETP/LIB
```

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

SATSSS74
LIS

SATSUT01
LIS

SATSUT05
LIS

SATSSS82
LIS

SATSSS90
LIS

SATSSS78
LIS

SATSSS80
LIS

SATSUT04
LIS

SATSUT06
LIS

SATSSS81
LIS

SATSSS91
LIS

SATSSS79
LIS

SATSSS83
LIS